FCA and pattern structures for mining care trajectories

Aleksey Buzmakov, Elias Egho, Nicolas Jay, Sergei O. Kuznetsov, Amedeo Napoli, Chedy Raïssi

FCA and pattern structures for mining care trajectories

Aleksey Buzmakov[1,2], Elias Egho[1], Nicolas Jay[1], Sergei O. Kuznetsov[2], Amedeo Napoli[1], and Chedy Raïssi[1]

[1] LORIA (CNRS – Inria NGE – U. de Lorraine), Vandœuvre-lès-Nancy, France
[2] National Research University Higher School of Economics, Moscow, Russia
{aleksey.buzmakov, chedy.raissi}@inria.fr,
{elias.egho, nicolas.jay, amedeo.napoli}@loria.fr, skuznetsov@hse.ru

Abstract. In this paper, we are interested in the analysis of sequential data and we propose an original framework based on Formal Concept Analysis (FCA). For that, we introduce sequential pattern structures, an original specification of pattern structures for dealing with sequential data. Pattern structures are used in FCA for dealing with complex data such as intervals or graphs. Here they are adapted to sequences. For that, we introduce a subsumption operation for sequence comparison, based on subsequence matching. Then, a projection, i.e. a kind of data reduction of sequential pattern structures, is suggested in order to increase the efficiency of the approach. Finally, we discuss an application to a dataset including patient trajectories (the motivation of this work), which is a sequential dataset and can be processed with the introduced framework. This research work provides a new and efficient extension of FCA to deal with complex (not binary) data, which can be an alternative to the analysis of sequential datasets.

Keywords: formal concept analysis, pattern structures, sequential pattern structures, sequences

Introduction

Sequence data is largely present and used in many applications. Consequently, mining sequential patterns from sequence data has become an important and crucial data mining task. In the last two decades, the main emphasis has been on developing efficient mining algorithms and effective pattern representations [1–5]. However, the problem with traditional sequential pattern mining algorithms (and generally with all pattern enumeration algorithms) is that they generate a large number of frequent sequences while few of them are truly relevant. Moreover, in some particular cases, only sequential patterns of a certain type are of interest and should be mined first. *Are we able to develop a framework for taking into account only patterns of required types?* In addition, another drawback for these pattern enumeration algorithms is that they depend on a user selected support threshold, which is usually hard to be properly set by non-experts. *How can one avoid the setting of support threshold while having optimal pattern analysis?*

The above questions can be answered by addressing the problem of analyzing sequential data with the formal concept analysis framework (FCA), an elegant

mathematical approach to data analysis [6], and pattern structures, an extension of FCA that handles complex data [7]. We explain the usage of projections which are mathematical functions respecting certain properties and allow to reduce the computational costs by reducing the volume of resulting patterns. Such a reduction helps an expert to interpret the extracted sequential patterns and reduces the famous "pattern flooding".

In this paper, we develop a novel and efficient approach for working with sequential pattern structures in FCA. The rest of the paper is organized as follows. Section 1 introduces main definitions of FCA and pattern structures. The next section defines sequential pattern structures. Then, before concluding the results are presented and discussed in Section 3.

1 FCA and Pattern Structures

FCA [6] is a mathematical formalism having many applications in data analysis. Pattern structures is a generalization of FCA for dealing with complex structures, such as sequences or graphs [7]. As it is a generalization it is enough to introduce only pattern structures.

Definition 1. *A pattern structure is a triple* $(G, (D, \sqcap), \delta)$*, where* G *is a set of objects,* $(D, \sqcap)$ *is a complete meet-semilattice of descriptions and* $\delta : G \to D$ *maps an object to a description.*

The lattice operation in the semilattice ($\sqcap$) corresponds to the similarity between two descriptions d_1 and d_2, i.e. the description which is common between d_1 and d_2. Standard FCA can be presented in terms of pattern structures in the following way. The set of objects G remains, while the semilattice of descriptions is $(\wp(M), \cap)$, where $\wp(M)$ is a powerset of M, and, thus, a description is a set of attributes, and the similarity operation corresponding to the set intersection, i.e. the similarity is the set of shared attributes. If $x = \{a, b, c\}$ and $y = \{a, c, d\}$ then $x \sqcap y = x \cap y = \{a, c\}$. The mapping $\delta : G \to \wp(M)$ is given by, $\delta(g) = \{m \in M \mid (g, m) \in I\}$, returning the describing set of attributes.

The Galois connection for a pattern structure $(G, (D, \sqcap), \delta)$ between the set of objects and the semilattice of descriptions is defined as follows:

$$A^\diamond := \prod_{g \in A} \delta(g), \qquad \text{for } A \subseteq G$$

$$d^\diamond := \{g \in G \mid d \sqsubseteq \delta(g)\}, \qquad \text{for } d \in D$$

Given a set of objects A, $A^\diamond$ returns the description which is common to all objects in A. And given a description d, $d^\diamond$ is the set of all objects whose description subsumes d. The partial order on D ($\sqsubseteq$) is defined w.r.t. the similarity operation $\sqcap$: $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$, and c is subsumed by d.

Definition 2. *A pattern concept of a pattern structure* $(G, (D, \sqcap), \delta)$ *is a pair* (A, d) *where* $A \subseteq G$ *and* $d \in D$ *such that* $A^\diamond = d$ *and* $d^\diamond = A$*,* A *is called the pattern extent and* d *is called the pattern intent.*

As in the standard case of FCA, a pattern concept corresponds to the maximal set of objects A whose description subsumes the description d, while there is no $e \in D$, subsuming d, i.e. $d \sqsubseteq e$, describing every object in A. The set of all concepts can be partially ordered w.r.t. partial order on extents (dually, intent patterns, i.e $\sqsubseteq$), within a concept lattice.

It is wroth mentioning, that the size of the concept lattice can be exponential w.r.t. to the number of objects, and, thus, we need a special ranking method to select the most interesting concepts for further analysis. Several such techniques are considered in [8], where it is shown that stability index [9] is more reliable in noisy data. Thus, we decided to use this index in the current work.

An example of pattern structures is given by Table 1a and described in the next sections, the corresponding lattice is depicted in Figure 1a.

2 Sequential Pattern Structures

2.1 An Example of Sequential Data

Patient	Trajectory
p^1	$\langle\{a\}\,;\{c,d\}\,;\{b,a\}\,;\{d\}\rangle$
p^2	$\langle\{c,d\}\,;\{b,d\}\,;\{a,d\}\rangle$
p^3	$\langle\{c,d\}\,;\{b\}\,;\{a\}\,;\{a,d\}\rangle$

(a) Sequential dataset.

	Subsequences		Subsequences
ss^1	$\langle\{c,d\}\,;\{b\}\,;\{d\}\rangle$	ss^5	$\langle\{a\}\,;\{d\}\rangle$
ss^2	$\langle\{d\}\,;\{a\}\rangle$	ss^6	$\langle\{a,d\}\rangle$
ss^3	$\langle\{c,d\}\,;\{b\}\rangle$	ss^7	$\langle\{a\}\rangle$
ss^4	$\langle\{c,d\}\,;\{b\}\,;\{a\}\rangle$		

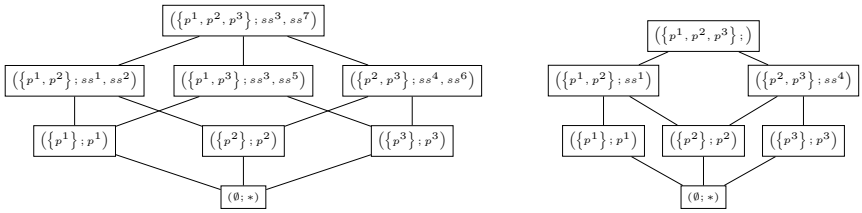
(b) Some common subsequences.

Table 1: A toy sequential dataset of patient medical trajectories.

A medical trajectory of a patient is a sequence of hospitalizations, where every hospitalization is described by a set of medical procedures the patient underwent. An example of medical trajectories for three patients is given in Table 1a. Patient p^1 had four hospitalizations and during the second hospitalization he underwent procedures c and d. Patients may have a different number of hospitalizations. Hereafter we use the following notation, different sequences are enumerated in superscript (p^1), while elements of a sequence are enumerated in the subscript ($p^1_2 = \{c,d\}$). One important task is to find the characteristic subsequences of hospitalizations for patients to optimize hospitalization processes. For example, we can find a strange sequence and, thus, motivate the deeper analysis of the problems behind or we can find typical sequences that allow us to estimate the treatment costs for a patient.

2.2 Partial Order on "Complex" Sequences

A sequence is constituted of elements from an alphabet. The classical subsequence matching task requires no special properties of the alphabet. Several generalization of the classical case were made by introducing subsequence relation based on itemset alphabet [10] or on multidimensional and multilevel alphabet [11]. Here, we generalize the previous cases, requiring for an alphabet



(a) Original Lattice. (b) Projected Lattice.

Fig. 1: The concept lattices for the pattern structure given by Table 1a. Concept intents refers to sequences in Tables 1a and 1b.

to form a semilattice $(E, \sqcap_E)$[1]. This generalization allows one to process in a unified way all types of complex sequential data.

Definition 3. *A sequence $t = \langle t_1; ...; t_k \rangle$ is a subsequence of a sequence $s = \langle s_1; ...; s_n \rangle$, denoted $t \leq s$, iff $k \leq n$ and there exist $j_1, ..j_k$ such that $1 \leq j_1 < j_2 < ... < j_k \leq n$ and for all $i \in \{1, 2, ..., k\}$, $t_i \sqsubseteq_E s_{j_i}$.*

With complex sequences and such kind of subsequences the calculation procedures can be difficult, thus, to simplify the procedure, only "restricted" subsequences are considered, where only the order of consequent elements is taken into account, i.e. given a j_1 in Definition 3, $j_i = j_{i-1} + 1$ for all $i \in \{2, 3, ..., k\}$. Such a restriction makes sens for our data, because a hospitalization is a discrete event and it is likely that the next hospitalization has a relation with the previous one, for example, hospitalizations for treating aftereffects of chemotherapy. Below the word "subsequence" refers to a "restricted" subsequence.

Based on Definition 3 and on the alphabet $(\wp(P), \cap)$, the sequence ss^1 in Table 1b is a subsequence of p^1 because if we set $j_i = i + 1$ (Definition 3) then $ss^1_1 \sqsubseteq p^1_2$ ($\{c, d\} \subseteq \{c, d\}$), $ss^1_2 \sqsubseteq p^1_3$ ($\{b\} \subseteq \{b, a\}$) and $ss^1_3 \sqsubseteq p^1_4$ ($\{d\} \subseteq \{d\}$).

2.3 Meet-semilattice of Sequences

Using the previous definitions, we can precisely define the sequential pattern structure. For that, we make an analogy with the pattern structures for graphs [12] where the meet-semilattice operation $\sqcap$ respects subgraph isomorphism. Thus, we introduce a sequential meet-semilattice respecting subsequence relation. Given an alphabet lattice $(E, \sqcap_E)$, D consists of sets of sequences based on E, such that if $d \in D$ contains a sequence s then all subsequences of s should be included into d, $\forall s \in d, \nexists \tilde{s} \leq s : \tilde{s} \notin d$. Similarity operation is the set intersection for two sets of sequences. Given two patterns $d_1, d_2 \in D$, the set intersection operation ensures that if a sequence s belongs to $d_1 \sqcap d_2$ then any subsequence of s belongs to $d_1 \sqcap d_2$ and thus $(d_1 \sqcap d_2) \in D$. As the set intersection operation is idempotent, commutative and associative, $(D, \sqcap)$ is a valid semilattice.

[1] In this paper we consider two semilattices, the first one is on the characters of the alphabet, $(E, \sqcap_E)$, and the second one is introduced by pattern structures, $(D, \sqcap)$.

However, the set of all possible subsequences can be rather large. Thus, it is more efficient and representable to keep a pattern $d \in D$ as a set of all maximal sequences $\tilde{d}$, $\tilde{d} = \{s \in d \mid \nexists s^* \in d : s^* \geq s\}$. Below, every pattern is given only by the set of maximal sequences. For example, $\{p^2\} \sqcap \{p^3\} = \{ss^4, ss^6\}$ (see Tables 1a and 1b), i.e. $\{ss^4, ss^6\}$ is the set of maximal common subsequences between $\{p^2\}$ and $\{p^3\}$, correspondingly $\{ss^4, ss^6\} \sqcap \{p^1\} = \{ss^3, ss^7\}$. Moreover, representing a pattern by the set of maximal sequences allows for an efficient implementation of the intersection "$\sqcap$" (see Section 3.1 for more details).

The sequential pattern structure for the example given by Subsection 2.1 is $(G, (D, \sqcap), \delta)$, where $G = \{p^1, p^2, p^3\}$, $(D, \sqcap)$ is the semilattice of sequential descriptions, and δ is the mapping associating an object in G to a description in D shown in Table 1a. Figure 1a shows the resulting lattice of sequential pattern concepts for this particular pattern structure $(G, (D, \sqcap), \delta)$.

2.4 Projections of Sequential Pattern Structures

Pattern structures can be hard to process due to the usually large number of concepts in the concept lattice and the complexity of the involved descriptions and the similarity operation. Moreover, a given pattern structure can produce a lattice with a lot of patterns which are not interesting for an expert. *Can we save computational time by deleting some unnecessary patterns?* Projections of pattern structures "simplify" to some degree the computation and allow one to work with a reduced description. In fact, projections can be considered as constraints (or filters) on patterns respecting certain mathematical properties. These mathematical properties guarantee that the projection of a lattice is a lattice where projected concepts have certain correspondence to original ones. We introduce projections on sequential patterns, adapting them from [7].

A projection $\psi : D \to D$ is an operator, which is monotone ($x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$), contractive ($\psi(x) \sqsubseteq x$) and idempotent ($\psi(\psi(x)) = \psi(x)$). A projection preserves the semilattice operation $\sqcap$ as follows. Under a projection ψ, the pattern structure $(G, (D, \sqcap), \delta)$ becomes the projected pattern structure $\psi((G, (D, \sqcap), \delta)) = (G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, where $D_\psi = \psi(D) = \{d \in D \mid \exists d^* \in D : \psi(d^*) = d\}$ and $\forall x, y \in D, x \sqcap_\psi y := \psi(x \sqcap y)$. The concepts of a projected pattern structure have a "similar" concept in the initial pattern structure [7].

One possible projection for sequential pattern structures comes from the following observation. In many cases it can be more interesting to analyze quite long common subsequences rather than small ones. For example, if we prefer common subsequences of length > 2, then between p^1 and p^2 in Table 1a there is only one maximal common subsequence, ss^1 in Table 1b, while ss^2 in Table 1b is too short to be considered as a common subsequence. Such kind of projections we call *Minimal Length Projection* (MLP) and depends on the parameter l, the minimal allowed length of the sequences in a pattern. The projected pattern concept lattice for MLP ≥ 3 is shown in Figure 1b. In the experimentation section we compare MLP projections with different value of the parameter. Projections are very useful, as they reduce the computational costs in a meaningful manner.

3 Sequential Pattern Structure Evaluation

3.1 Implementation

Nearly all state-of-the-art FCA algorithms can be adapted to process pattern structures. We adapted `AddIntent` algorithm [13], as the lattice structure is important for us to calculate stability (see the algorithm for calculating stability in [14]). To compute the semilattice operation $(\sqcap, \sqsubseteq)$ between two sets of sequences $S = \{s^1, ...s^n\}$ and $T = \{t^1, ..., t^m\}$, $S \sqcap T$ is calculated according to Section 2.3, i.e. maximal sequences among all maximal common subsequences for any pair of s^i and t^j. To find all common subsequences of two sequences, the following observations is useful. If $ss = \langle ss_1; ...; ss_l \rangle$ is a subsequence of $s = \langle s_1; ...; s_n \rangle$ with $j_i^s = k^s + i$ (Definition 3: k^s is the index difference from which ss is a subsequence of s) and a subsequence of $t = \langle t_1; ...; t_m \rangle$ with $j_i^t = k^t + i$ (likewise), then for any index $i \in \{1, 2, ..., l\}$, $ss_i \sqsubseteq_E (s_{j_i^s} \sqcap t_{j_i^t})$. Thus, to find maximal common subsequences between s and t, we, first, align s and t in all possible ways, and then we compute the resulting intersections and keep only the maximal ones. Let us consider two possible alignments of s^1 and s^2:

$$
\begin{array}{rl}
s^1 = & \langle \{a\}\,; \{c,d\}\,; \{b,a\};\ \{d\}\ \rangle \\
s^2 = & \langle \{c,d\}; \{b,d\}\,; \{a,d\} \rangle \\
ss^l = & \langle\ \ \emptyset\ \ ;\ \{d\}\ \rangle
\end{array}
\qquad
\begin{array}{rl}
s^1 = & \langle \{a\}\,; \{c,d\}; \{b,a\};\ \{d\}\ \rangle \\
s^2 = & \langle \{c,d\}; \{b,d\}; \{a,d\} \rangle \\
ss^r = & \langle \{c,d\};\ \{b\}\ ;\ \{d\}\ \rangle
\end{array}
$$

The left intersection ss^l is smaller than ss^r and, thus, is not kept.

3.2 Experiments and Discussion

The experiments are carried out on an "Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz" computer with 8Gb of memory under the Ubuntu 12.04 operating system. The algorithms are not parallelized and are coded in C++.

First, the public available dataset from UCI repository on anonymous web data is used as a benchmark data set for scalability tests. This dataset contains around 10^6 transactions, and each transaction is a sequence based on "simple" alphabet, i.e. with no order on the elements. The overall time changes from 37279 seconds for the sequences of length $MLP \geq 5$ upto 97042 seconds for the sequences of length $MLP \geq 3$. For more details see the web-page.[2]

Our use-case data set comes from a French healthcare system [15]. In the experiment, 1000 patients are analyzed. The dataset describes a patient as a sequence of hospitalizations without any timestamps. A hospitalization for a patient is a tuple with the hospital name, the cause for the hospitalization and the set of procedures the patient underwent. All the field of a hospitalization are joined into one single set, which describes one element of a sequence.

Table 2 shows the final times and the lattice sizes for different projections. The calculation time with projections changes from 3120 to 4510 seconds depending on the projection parameter. The lattice sizes for the different projections

[2] `http://www.loria.fr/~abuzmako/FCA4AI2013/experiment-uci.html`

Projection	No	$l=2$	$l=3$	$l=4$	$l=5$	$l=6$
Time(s)	$> 1.4 * 10^5$	4510	3878	3722	3435	3120
Lattice	$> 2.8 * 10^6$	554332	271166	189353	137912	100492

Table 2: The processing time and the lattice size for the PMT dataset.

	Intent	Extent
c_1	$\langle \{D_{\text{Chemotherapy}}\} * 8 \rangle$	284 (28%)
c_2	$\langle \{D_{\text{Device Adj.}}, P_{\text{Artery Catheter}}\}; \{D_{\text{Chemotherapy}}\} * 12 \rangle$	74 (7%)
c_3	$\langle \{P_{\text{Chest Radiography}}\} * 2 \rangle$	189 (19%)

Table 3: Some interesting concepts of the PMT database

changes from 10^5 to $5 \cdot 10^5$. Notice that the ratio for the lattice size is of 5 while the ratio for computation time is 1.5. Deletion of short sequences can dramatically change the lattice size, since the shorter a sequence is, the more probable it is a subsequence of a patient trajectory, but the computation of the semilattice operation is easily processed with shorter sequences. The calculation without projection takes a lot of time with relatively small lattice size (40 times more for the runtime and 6 times more for the lattice size w.r.t the projection $l = 2$). The reason for that is memory swapping to the hard disk. Thus, the best projection for that dataset is $l = 2$ as its computation time is reasonable and it preserves the most of the information among the other projections.

Table 3 shows some interesting concept intents and the sizes of the corresponding extents. The concept are selected among the most stable ones [9]. The concept c_1 corresponds to 28% of the patients having at least 8 consequent hospitalizations because of chemotherapy. Now we can estimate the minimal cost of the overall procedures for a patient (we know the price of chemotherapy and we know the expected number of procedures). Concept c_2 covers 7% of the patients, and its intent is more interesting: 12 hospitalizations for chemotherapy following the hospitalization for adjustment of a chemotherapy device and an artery catheter installation. Both concepts c_1 and c_2 can be found within any of the considered projections (with $l \in \{2, 3, .., 6\}$), while the concept c_3 can be found only within the most specific projection ($l = 2$). The concept c_3 covers 19% of the patients and describes patients with at least two consequent hospitalizations accompanied with a chest radiography. In this way, we have a kind of control of the trajectory quality, because we could have an under-examination of a patient during the first hospitalization.

Conclusion

In this paper, we present an approach for analyzing sequential datasets within the framework of pattern structures, an extension of FCA dealing with complex (non binary) data. Using pattern structures leads to the construction of a pattern concept lattice, which does not require the setting of a support threshold, as usually needed in sequential pattern mining. Another point worth to be noticed is that the use of projections gives a lot of flexibility especially for mining and interpreting special kinds of patterns. The framework is tested on a real-life

dataset recording patient trajectories in a healthcare system. Interesting patterns are extracted and then interpreted, showing the feasibility and usefulness of the approach. For the future work, it is important to more deeply investigate projections, their potentialities w.r.t. the types of patterns.

Acknowledgements: this research received funding from the Basic Research Program at the National Research University Higher School of Economics (Russia) and from the BioIntelligence project (France).

References

1. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.: FreeSpan: frequent pattern-projected sequential pattern mining. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. (2000) 355–359
2. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan Mining Sequential Patterns Efficiently by Prefix Projected Pattern Growth. In: 17th International Conference on Data Engineering. (2001) 215–226
3. Yan, X., Han, J., Afshar, R.: CloSpan: Mining Closed Sequential Patterns in Large Databases. In: In SDM. (2003) 166–177
4. Ding, B., Lo, D., Han, J., Khoo, S.C.: Efficient Mining of Closed Repetitive Gapped Subsequences from a Sequence Database. In: 2009 IEEE 25th International Conference on Data Engineering, IEEE (March 2009) 1024–1035
5. Raïssi, C., Calders, T., Poncelet, P.: Mining conjunctive sequential patterns. Data Min. Knowl. Discov. **17**(1) (2008) 77–93
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer, Secaucus, NJ, USA (1997)
7. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In Delugach, H., Stumme, G., eds.: Conceptual Structures: Broadening the Base SE - 10. Volume 2120 of LNCS. Springer Berlin Heidelberg (2001) 129–142
8. Klimushkin, M., Obiedkov, S.A., Roth, C.: Approaches to the Selection of Relevant Concepts in the Case of Noisy Data. In: Proceedings of the 8th international conference on Formal Concept Analysis. ICFCA'10, Springer (2010) 255–266
9. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4) (2007) 101–115
10. Casas-Garriga, G.: Summarizing Sequential Data with Closed Partial Orders. In: SDM. (2005)
11. Plantevit, M., Laurent, A., Laurent, D., Teisseire, M., Choong, Y.W.: Mining multidimensional and multilevel sequential patterns. ACM Transactions on Knowledge Discovery from Data **4**(1) (January 2010) 1–37
12. Kuznetsov, S.: Learning of Simple Conceptual Graphs from Positive and Negative Examples. Volume 1704 of LNCS. Springer (1999) 384–391
13. Merwe, D.V.D., Obiedkov, S., Kourie, D.: AddIntent: A new incremental algorithm for constructing concept lattices. In Goos, G., Hartmanis, J., Leeuwen, J., Eklund, P., eds.: Concept Lattices. Volume 2961. Springer (2004) 372–385
14. Roth, C., Obiedkov, S., Kourie, D.G.: On succinct representation of knowledge community taxonomies with formal concept analysis A Formal Concept Analysis Approach in Applied Epistemology. International Journal of Foundations of Computer Science **19**(02) (April 2008) 383–404
15. Fetter, R.B., Shin, Y., Freeman, J.L., Averill, R.F., Thompson, J.D.: Case mix definition by diagnosis-related groups. Med Care **18**(2) (February 1980) 1–53